

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されて
いる事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed
with this Office.

出願年月日 2003年 9月 1日
Date of Application:

出願番号 特願 2003-309293
Application Number:

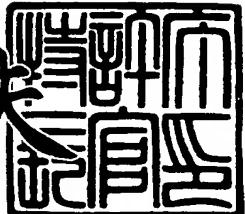
[ST. 10/C] : [JP 2003-309293]

出願人 ミノルタ株式会社
Applicant(s):

2004年 1月 20日

特許庁長官
Commissioner,
Japan Patent Office

今井康夫



【書類名】 特許願
【整理番号】 IT01051
【提出日】 平成15年 9月 1日
【あて先】 特許庁長官 殿
【国際特許分類】 H04N 1/00
G06F 3/12
G06F 9/44

【発明者】
【住所又は居所】 大阪府大阪市中央区安土町二丁目3番13号 大阪国際ビル ミノルタ株式会社内
【氏名】 柴田 浩一

【発明者】
【住所又は居所】 大阪府大阪市中央区安土町二丁目3番13号 大阪国際ビル ミノルタ株式会社内
【氏名】 村上 正和

【特許出願人】
【識別番号】 000006079
【氏名又は名称】 ミノルタ株式会社

【代理人】
【識別番号】 100090446
【弁理士】
【氏名又は名称】 中島 司朗

【手数料の表示】
【予納台帳番号】 014823
【納付金額】 21,000円

【提出物件の目録】
【物件名】 特許請求の範囲 1
【物件名】 明細書 1
【物件名】 図面 1
【物件名】 要約書 1
【包括委任状番号】 9716120

【書類名】特許請求の範囲**【請求項 1】**

画像形成部、読み取り部、表示部の少なくとも一つを有するハードウェア資源と、前記ハードウェア資源の上位に位置する第1の制御プログラムと、前記第1の制御プログラムの上位に位置するアプリケーションおよび第2の制御プログラムと、からなるアーキテクチャを有し、

前記第1の制御プログラムは、

予め定義された関数により前記アプリケーションおよび前記第2の制御プログラムからの画像処理に関する処理要求を受け付ける第1のAPI（アプリケーション・プログラム・インターフェース）を有し、受け付けた処理要求に基づく処理を前記ハードウェア資源に実行させ、

前記第2の制御プログラムは、

予め外部に公開された関数により前記外部から画像処理に関する処理要求を受け付ける第2のAPIを有し、受け付けた処理要求を前記第1のAPIに対応するコマンドに変換して前記第1の制御プログラムに渡すことを特徴とする画像処理装置。

【請求項 2】

前記第1の制御プログラムは、

前記第2の制御プログラムからの処理要求が、前記アプリケーションに対するものである場合には、当該処理要求を前記アプリケーションに渡すことを特徴とする請求項1に記載の画像処理装置。

【請求項 3】

前記外部からの画像処理に関する処理要求は、XMLで表されたものであり、

前記第2の制御プログラムは、

前記外部からの画像処理に関する処理要求をXML形式のデータで受け付けることを特徴とする請求項1もしくは2に記載の画像処理装置。

【請求項 4】

前記第2の制御プログラムは、

受け付けたXML形式のデータから前記処理要求を示す情報を抽出する第1の変換部と

抽出された情報を前記第1のAPIに対応するコマンドに変換する第2の変換部と、を備えることを特徴とする請求項3に記載の画像処理装置。

【書類名】明細書

【発明の名称】画像処理装置

【技術分野】

【0001】

本発明は、画像処理装置に関する。

【背景技術】

【0002】

従来、会社のオフィス等では、セットされた原稿の画像を読み取るスキャンジョブ、原稿の画像を読み取って得られた画像データに基づいて用紙に印刷するコピージョブ、外部端末からの印刷指示に基づいて用紙に印刷するプリントジョブ、外部のファクシミリ装置との通信により画像を送受信するFAXジョブなどの複数の機能を有する、いわゆるデジタル複合機（MFP：Multiple Function Peripheral。以下、「MFP」という。）が普及している。

【0003】

このようなMFPを用いれば、ユーザにとっては、コピー機、プリンタ等の装置を個別に配置する必要がなく、1台で複数の機能を実行できるため、低コストであり使い勝手も良くなる。

一方で、MFPの開発者にとっては、スキャン機能等の各種機能に対応する複数のソフトウェア（アプリケーション。以下、「アプリ」という。）を開発する必要があるが、これらを個別に設ける構成としたのでは開発に多大な時間がかかってしまう。そのため、例えば特許文献1には、各アプリの共通部分を括り出したものと汎用OS（オペレーティング・システム）上でプラットフォームを形成し、アプリ開発時には当該プラットフォーム以外の部分を作成するだけの構成とすることで、効率良く開発を行えるようにした技術が開示されている。

【特許文献1】特開2002-84383号公報

【発明の開示】

【発明が解決しようとする課題】

【0004】

上記公報記載の構成とすれば、開発段階においては効率が良くなるものの、ユーザへの納品後に、ユーザから、現存のアプリには備えられていない機能を実行したい旨の要求（いわゆるカスタマイズの要求）を受けた場合には容易に対応できない。例えば、スキャン機能について、現在のスキャンアプリにより提供されている機能以外の読み取りモードで読み取り動作を行いたいような場合には、現存のアプリをそのユーザのためだけに全面的に変更する必要があり、大変な作業を強いられ、実際には困難である。すなわち、機能拡張性に極めて乏しい構成になっている。

【0005】

本発明は、上述のような問題点に鑑みてなされたものであって、機能を容易に拡張でき、その場合に現存のアプリの変更をほとんど必要としない構成の画像処理装置を提供することを目的とする。

【課題を解決するための手段】

【0006】

上記目的を達成するため、本発明に係る画像処理装置は、画像形成部、読み取り部、表示部の少なくとも一つを有するハードウェア資源と、前記ハードウェア資源の上位に位置する第1の制御プログラムと、前記第1の制御プログラムの上位に位置するアプリケーションおよび第2の制御プログラムと、からなるアーキテクチャを有し、前記第1の制御プログラムは、予め定義された関数により前記アプリケーションおよび前記第2の制御プログラムからの画像処理に関する処理要求を受け付ける第1のAPI（アプリケーション・プログラム・インターフェース）を有し、受け付けた処理要求に基づく処理を前記ハードウェア資源に実行させ、前記第2の制御プログラムは、予め外部に公開された関数により前記外部から画像処理に関する処理要求を受け付ける第2のAPIを有し、受け付けた処

理要求を前記第1のA P Iに対応するコマンドに変換して前記第1の制御プログラムに渡すことを特徴とする。

【0007】

また、前記第1の制御プログラムは、前記第2の制御プログラムからの処理要求が、前記アプリケーションに対するものである場合には、当該処理要求を前記アプリケーションに渡すことを特徴とする。

さらに、前記外部からの画像処理に関する処理要求は、XMLで表されたものであり、前記第2の制御プログラムは、前記外部からの画像処理に関する処理要求をXML形式のデータで受け付けることを特徴とする。

【0008】

また、前記第2の制御プログラムは、受け付けたXML形式のデータから前記処理要求を示す情報を抽出する第1の変換部と、抽出された情報を前記第1のA P Iに対応するコマンドに変換する第2の変換部と、を備えることを特徴とする。

【発明の効果】

【0009】

本発明に係る画像処理装置は、第2の制御プログラムを有しているので、例えば当該装置がユーザに納品された後であっても、現存のアプリケーションに変更を加えることなく、当該アプリケーションでは実行できなかった新たな機能を第2の制御プログラムを利用して外部から実行させることができになり、機能拡張性が向上し、ユーザにとって使い勝手が大変良くなる。

【0010】

そして、ハードウェア資源、第1の制御プログラム、アプリケーションがこの順に階層構造的に体系化されてなる従来の構成に、第1の制御プログラムから見てアプリケーションと同層の位置に第2の制御プログラムを配すれば良く、基本的に現存のアプリケーションをそのまま使用できる。これにより、画像処理装置の開発者側にとっては、第2の制御プログラムを配置する場合に、アプリケーションの変更、確認作業等に手をとられる必要がなく、開発期間を大幅に短縮できると共に開発負担を軽減できる。

【0011】

また、外部からの処理要求がアプリケーションへのものである場合には、当該処理要求が当該アプリケーションに渡されるようにしている。

これにより、例えば外部から当該アプリケーションを起動させることができになり、いわゆるリモート機能を実現できる。

さらに、外部からの画像処理に関する処理要求がXML形式のデータで受け付けられる構成になっている。

【0012】

これにより、外部ユーザは、画像処理装置に実行させたいジョブの画像処理に関する処理要求をXMLを用いて記述することができ大変扱い易い。

また、画像処理装置が第1の変換部を備える構成になっている。

これにより、外部と画像処理装置との間の通信プロトコルとして、例えば周知のS O A P等を用いることができ、外部ユーザによって簡易かつ扱い易い。

【発明を実施するための最良の形態】

【0013】

以下、本発明に係る画像処理装置としてのM F Pの実施の形態について図面を参照しながら説明する。

(1) M F Pの構成

図1は、M F P 1のシステム構成を示す図である。

同図に示すように、M F P 1は、最下位層に位置するデバイス（ハードウェア資源）10と、デバイス10を直接制御する制御層20と、制御層20の上位に位置するアプリケーション層30が階層構造的に体系化されたアーキテクチャからなり、上記スキャンジョブ、コピージョブ、プリントジョブ、F A Xジョブなどの画像処理ジョブを実行する複

数の機能を有するデジタル複合機である。また、外部機器、ここではPC（パーソナルコンピュータ）50とLAN等のネットワーク（不図示）を介して接続されており、例えばTCP/IP(Transmission Control Protocol/Internet Protocol)を用いて互いに各種データのやりとりが可能になっている。

【0014】

デバイス10には、画像形成部としてのプリンタ11と、読み取り部としてのスキャナ12と、操作パネル13およびI/O14が含まれている。

ここで、プリンタ11は、周知の電子写真方式により画像データに基づいて画像を用紙に印刷（プリント）するものである。スキャナ12は、セットされた原稿の画像をCCD（Charge Coupled Device）等の光電変換素子により読み取って画像データを得る公知の装置である。操作パネル13は、入力キーや表示部としてのディスプレイ等を備え、操作者からのキー入力を受け付けると共に各種メッセージ等をディスプレイに表示させるものである。また、I/O14は、CPU、RAM、ROM、ハードディスク、外部機器とのインターフェース装置（例えば、NIC：Network Interface Card）等を含むものである。

【0015】

制御層20は、カーネル21と、各デバイスの動作を制御するためのモジュール（IJC：Image Job Controller）22と、外部機器との通信制御を実行するためのインターフェースモジュール（IO：Input-Output）23と、IO API、IJC API（以下、これらをまとめて「IO/IJC API」という。）24とを備えており、OS（Operating System）として機能する。

【0016】

ここで、IO/IJC API24は、アプリケーション層30の各アプリケーション31～36にIJC22とIO23の機能（すなわち、制御層20の機能）を提供するためのAPI（アプリケーション・プログラム・インターフェース）であり、各アプリがこの機能を利用する際に呼び出すための関数の集まりである。すなわち、予め定義されたこれら関数により各アプリからの画像処理に関する処理要求を受け付ける機能を有するものである。

【0017】

アプリケーション層30には、IO/IJC API24にアクセスして（予め定義された関数の内の任意のものを必要に応じて呼び出して（コールして））、実際にデバイスを動作させるための複数のアプリケーションが含まれている。

例えば、パネルアプリ31は、操作パネル13のディスプレイ表示を制御するためのアプリケーションである。具体的には、操作パネル13からのキー入力情報を制御層20を介して受け付けると、その入力情報に基づくメッセージ等を操作パネル13のディスプレイに表示させる。例えば、コピースタートキーが押されたことを示す情報を受け付けると、「コピー中です」というメッセージを表示させる。

【0018】

プリントアプリ32は、プリントジョブ実行時に起動されると、ここでは外部機器からのPDL（Page Description Language）形式のプリントデータをビットマップ形式の画像データに変換し、その画像データに基づくプリント処理をプリンタ11に実行させる。

スキャンアプリ33は、スキャンジョブ実行時に起動されると、スキャナ12に原稿画像を読み取らせ（スキャン動作を実行させて）、画像データを得る。

【0019】

コピーアプリ34は、コピージョブ実行時に起動されると、スキャナ12に原稿画像を読み取らせ、得られた画像データに基づくプリント処理をプリンタ11に実行させる。

FAXアプリ35は、FAXジョブ実行時に起動されると、FAX送信時には、例えばスキャナ12に原稿画像を読み取らせ、得られた画像データをファクシミリ用のデータに変換して、指定された宛先に送信し、FAX受信時には、例えば外部から受信した画像データに基づくプリント処理をプリンタ11に実行させる。

【0020】

外部APIアプリ36は、外部機器からの画像処理に関する処理要求に基づいて、デバイスの動作を制御するプログラムであり、外部API361、XML変換部362およびAPI変換部363を備えている。

外部API361は、外部に制御層20の機能を提供するためのAPIであり、PC50が制御層20の機能を利用する際に呼び出すための関数の集まりである。すなわち、予め定義されたこれら関数により外部機器からの画像処理に関する処理要求を受け付ける機能を有するものである。

【0021】

この外部API361は、予め一般の外部ユーザ（例えば、ソフトウェア開発者等）に公開されるようになっており、外部ユーザは、この公開されたAPIに基づいて制御層20の機能を利用するためのソフトウェアを開発することができる。

具体的には、外部ユーザは、公開されたAPIから、MFP1のデバイス制御のために、どのようなコマンドやパラメータ類（例えば、実行させるべきジョブの指定、ジョブ開始の指示等を示すコマンド、スキヤンジョブの場合の読み取りサイズ、解像度等、コピージョブの場合のコピー枚数等のジョブ実行条件を示すパラメータ等）が用意されており、そのコマンド等をどのような構文で記述すれば良いのかを知ることができる。

【0022】

外部ユーザは、当該コマンド等を当該構文に基づいて記述したデータ（ここでは、XML（Extensible Markup Language）形式のデータ。図2参照）を画像処理に関する処理要求を含むデータとしてMFP1に送信させるためのソフトウェアを開発し、当該ソフトウェアを用いて上記XMLデータをMFP1に送信させるようにすることで、所望のデバイスに所望の動作をさせることができるものである。

【0023】

外部API361は、IO/IJC API24で定義されている関数群の中の、所定の複数のものを一つにまとめた形態、例えばデバイスのある動作が、IO/IJC API24で関数A、B、Cにより定義されている場合、関数A、B、Cをまとめた関数D（すなわち、関数Dは、関数A、B、Cを実行させるための関数ということになる。）で構成されている。

【0024】

このようにまとめたある動作を一の関数で表すことにより、外部ユーザは、例えばデバイスに当該ある動作を実行させたい場合、PC50から関数Dを呼び出すようにすれば、実質的に関数A、B、Cを呼び出したことになり、複数のコマンドを用いる必要がなくなりて扱い易く、上記ソフトウェアも開発し易くなる。なお、一の動作を一の関数で表したものについては、その関数が用いられる。

【0025】

XML変換部362は、後述のように、PC50から送られて来るXMLデータから処理要求を示す情報としてのコマンド等を抽出する。

API変換部363は、XML変換部362で抽出されたコマンド等（PC50からの画像処理に関する処理要求）を、制御層20で実行できる所定のデータ形式に変換して制御層20に渡す。なお、PC50と外部APIアプリ36間でのXMLデータの通信プロトコルとして、ここでは周知のSOAP（Simple Object Access Protocol）が用いられる。

【0026】

一方、PC50は、図示していないが、PC本体、ディスプレイ、キーボード、ネットワークインターフェースおよびSDK（Software Development Kit）51等を備えている。このSDK51は、上記ソフトウェアを作成するためのものとして外部ユーザに利用されるものである。

なお、ホストPC側とMFP側の境界位置に破線で示した「外部API（XML）」は、PC50により利用される外部APIがXMLで記述されたデータで扱われていること

を示したものである。

【0027】

上記のような外部A P I アプリ36を備える構成により、M F P 1の各アプリでは行えなかった機能であっても、外部から実行させることが可能になる。

例えば、スキャナ12自体は、複数の読み取り解像度、例えば1200、600、400 d p i (dot per inch) のいずれの解像度でも原稿画像を読み取る能力を有しているが、スキャンアプリ33には、ユーザからの解像度選択入力を受け付ける機能が備えられていない（例えば、最低の400 d p i 固定になっている）構成の場合に、外部A P I 361として読み取り解像度を指定したスキャンジョブの実行に必要な内容を公開しておけば、外部ユーザは、当該スキャンジョブを実行させるために必要なコマンド等を含むX M LデータをM F P 1に送る処理を行うソフトウェアを外部で作成でき、外部から当該スキャンジョブを実行させることができる。

【0028】

以下、図2、3を用いて当該スキャンジョブを実行する場合の具体例を説明する。

図2は、公開されたA P Iに基づいて外部ユーザにより作成された、上記スキャンジョブを実行させるためのコマンド等を含むX M Lデータが、P C 50からM F P 1の外部A P I アプリ36を介して制御層20に渡されるまでに変換されて行く様子を例示した模式図である。

【0029】

ここでは、原稿枚数「1枚」、原稿サイズ「A4」、原稿のセット方向「ポートレート（たて長）」、「カラー原稿」、読み取り解像度「600（d p i）」、「J p e g高圧縮」、ファイル名「t e s t f i l e」、読み取った画像データの送信宛先（P C 50）を「a b c @m i n o l t a . c o . j p」としたジョブを実行させる場合の例を示している。なお、同図はX M Lデータの構成を解り易いように説明に必要な部分だけを簡素化して示した例であって、実際には一のジョブを実行させるために必要となるコマンド等の数、種類はこれよりも多くなり、また通信データとしてはヘッダ等の各種情報が付加されることになる。

【0030】

同図のブロック1に示すように、P C 50からは、上記スキャンジョブの原稿サイズ等の各項目の内容が所定のタグで挟まれた状態で記述されたX M Lデータが出力される。ここでは、「message ID」タグに挟まれた「scan」がコマンドに相当し、「parameter」タグに挟まれた「destination」等が当該コマンドに対応するパラメータとなる。

このX M Lデータは、M F P 1のI O 2 3を介して外部A P I アプリ36のX M L変換部362に送られ、ブロック2に示すデータに変換される。

【0031】

具体的には、X M L変換部362において、入力されたX M Lデータからコマンド等に相当する部分と、それに対応するパラメータとが処理要求を示す情報として抽出される。この意味で、ブロック2に示すデータが、外部A P I 361として公開された関数「scan」がP C 50により呼び出された場合のデータ構成例となる。

ブロック2のデータは、A P I変換部363に送られ、制御層20で直接実行できる形態（すなわち、I O / I J C A P I 2 4に対応するコマンド）に変換される（ブロック3）。

【0032】

このブロック3のデータの「do_scan」「scan_destination」は、制御層20で実際にスキャン動作を実行するための関数を示しており、その後の括弧内のパラメータは、その関数の引数になっている。

制御層20では、上記関数が呼び出されるとそれらを実行する。具体的には、設定された読み取り条件（原稿A4サイズ、読み取り解像度600 d p i等）に基づいてスキャナ11に原稿画像の読み取り動作を実行させ、読み取られた画像データを、設定された送信宛先としてのP C 50（a b c @m i n o l t a . c o . j p）に送信させる。

【0033】

図3は、上記スキャンジョブを実行する場合におけるPC50とMFP1間のデータフローを示す模式図である。ここで同図の矢印の内、細線は、IO/IJC API24がコールされることにより実行される動作を示し、太線は、外部API361がコールされることにより実行される動作を示したものであり、二重線は画像データを示している。このことは、図4、5についても同様である。

【0034】

外部APIアプリ36は、PC50からスキャン開始のための外部APIコールを受けると(S1)、PC50とXMLデータ(図2のブロック1に示す読み取り解像度等を含むデータ)を送受信する(S2)。

外部APIアプリ36は、制御層20に対しへスキャン動作指示を行い(S3)、その指示に対する結果、例えばスキャン開始の承諾等を受け付けると(S4)、その旨をPC50に通知する(S5)。そして、制御層20からジョブ開始可能な状態になっていることを受け付けると(S6)、その旨をPC50に通知し(S7)、制御層20に読み取り解像度、原稿サイズ等の読み取り条件、送信宛先等を通知する(S8)。

【0035】

制御層20は、読み取り解像度等の読み取り条件に基づいてスキャン動作を開始させ、IJC22を介して画像データを外部APIアプリ36に送る(S9)。

外部APIアプリ36は、制御層20(IJC22)から画像データを取得すると、当該画像データをIO23を介して送信宛先としてのPC50に送信させる(S10)。

外部APIアプリ36は、その後ジョブ終了を受け付けると(S11)、その旨をPC50に通知して(S12)、処理を終了する。

【0036】

このように、公開されたAPIに基づいてスキャンジョブに必要なコマンド、パラメータをPC50から送信することで、スキャンアプリ33を起動させなくてもスキャンアプリ33では実行できない機能を外部から実行させることが可能になる。

上記では、スキャンアプリ33を起動させない場合の例を説明したが、スキャンアプリ33を起動させてスキャン動作を行うスキャンジョブの実行に必要な内容を外部API361として公開しておけば、外部から当該スキャンジョブを実行させることも可能である。以下、図4を用いて当該スキャンジョブを実行させる場合の例を説明する。

【0037】

図4は、スキャンアプリ33を起動させてスキャンジョブを実行させる場合のデータフローを示した模式図である。

外部APIアプリ36は、IO23を介し、PC50からスキャン開始のための外部APIコールを受けると(スキャンアプリ33を起動させてスキャン動作を実行させるための所定のコマンドを受け付けると)(S21)、PC50とXMLデータ(原稿サイズ等の読み取り条件、送信宛先を示すデータ)を送受信し(S22)、スキャンアプリ33へ接続確認を依頼する(S23)。スキャンアプリ33とIO23との間で接続確認がなされ(S24)、接続確認結果が通知されると(S25)、外部APIアプリ36は、制御層20からその通知を取得する(S26)。

【0038】

外部APIアプリ36は、スキャン動作指示を制御層20に行い(S27)、その指示に対する結果、例えばスキャン開始の承諾等を受け付けると(S28)、その旨をPC50に通知する(S29)。そして、ジョブ開始可能状態であることを受け付けると(S30)、その旨をPC50に通知し(S31)、スキャンアプリ33に原稿サイズ等の読み取り条件、送信宛先を通知する(S32)。

【0039】

スキャンアプリ33は、原稿サイズ等の読み取り条件に基づいてスキャン動作を開始させると共に画像データ取得のための要求を行い(S33)、制御層20(IJC22)から画像データを取得すると(S34)、当該画像データに所定の圧縮等の処理を施して、

その画像データを上記送信宛先に I O 2 3 を介して送信させる (S 3 5)。

外部 A P I アプリ 3 6 は、ジョブの実行状態、例えば送信のためのファイル変換中、送信中等の情報を受け付け (S 3 6) 、その後ジョブ終了を受け付けると (S 3 7) 、その旨を P C 5 0 に通知し (S 3 8) 、処理を終了する。

【0040】

このようにスキャンアプリ 3 3 を起動させてスキャン動作を実行させる場合には、外部からアプリ起動のためのコマンドを送信すれば、スキャンアプリ 2 3 がスキャナ 1 2 を制御してスキャン動作を実行してくれるので、外部ユーザにとってみれば、原稿画像の読み取り開始から、読み取られた画像データを設定された送信あて先に送信させるまでの一連の動作をいちいち所定のコマンド等を用いて指示する必要がなく、その意味でスキャンジョブを簡易に外部から実行させることが可能になる。

【0041】

次に図 5 を用いて、プリントアプリ 3 2 を起動させないで、外部からプリントジョブを実行させる場合の例を説明する。この場合も上記同様に、プリントアプリ 3 2 を起動させないでプリントジョブを実行させる場合における当該プリントジョブの実行に必要な内容が予め外部 A P I 3 6 1 として公開されることになる。

図 5 に示すように、外部 A P I アプリ 3 6 は、P C 5 0 からプリント開始のための外部 A P I コールを受けると (プリントアプリ 3 2 を起動させずにプリント動作を実行させるための所定のコマンドを受け付けると) (S 4 1) 、P C 5 0 から画像データを取得する (S 4 2) 。そして、P C 5 0 と X M L データ (例えば、プリント条件としての用紙サイズ、プリント枚数等を示すデータ) を送受信する (S 4 3) 。

【0042】

外部 A P I アプリ 3 6 は、制御層 2 0 の I J C 2 2 に、プリント指示、必要な設定情報として用紙サイズ等のプリント条件を通知し (S 4 4) 、画像データを送信する (S 4 5) 。

制御層 2 0 は、プリンタ 1 1 を制御して、受信したプリント条件、画像データに基づくプリント動作を開始させる。また、ジョブの実行状態、例えばプリント中、終了等の情報を外部 A P I アプリ 3 6 に送る (S 4 6) 。

【0043】

外部 A P I アプリ 3 6 は、制御層 2 0 からプリント中を示す情報を受け付け、その後プリント終了を示す情報を受け付けると (S 4 7) 、ジョブ終了である旨を P C 5 0 に通知して (S 4 8) 、処理を終了する。

このように、P C 5 0 から、プリンタ 1 1 にプリントを実行させるためのコマンド等を送信すれば、プリントアプリ 3 2 を起動させなくても、プリントアプリ 3 2 では扱えないデータ、例えばビットマップ形式のデータを画像データとして直接 P C 5 0 から M F P 1 に送信させてプリントを実行せざるが可となり、P C 5 0 側で扱えるデータがプリントアプリ 3 2 により制限されず、その意味で機能拡張性が向上する。

【0044】

なお、図 2 では、スキャンジョブの場合におけるデータ構成例を説明したが、他のプリントジョブ、コピージョブ等の場合も同様に、ジョブを実行させるためのコマンド、パラメータ等から構成される X M L 形式のデータが、画像処理に関する処理要求を含むデータとして、P C 5 0 から M F P 1 に送られ、外部 A P I アプリ 3 6 において制御層 2 0 に対応するコマンドに変換されて制御層 2 0 に渡されることになる。

【0045】

以上説明したように、本実施の形態の M F P 1 は、外部からの、画像処理に関する処理要求を含む X M L データを受け付けて、それを制御層 2 0 (第 1 の制御プログラム) の A P I に対応するコマンドに変換して制御層 2 0 に渡す機能を有する外部 A P I アプリ 3 6 (第 2 の制御プログラム) を備えているので、M F P 1 がユーザに納品された後であっても、現存のアプリに変更を加えることなく、これらアプリでは実行できなかつた新たな機能、例えば上記のように外部から読み取り解像度を指定してスキャンジョブを実行させる

機能、プリントアプリ32では扱えないデータを画像データとして送信してプリントジョブを実行させる機能等を実行させることができになり、機能拡張性が向上し、ユーザにとって使い勝手が大変良くなる。

【0046】

また、ハードウェア資源10、制御層20、アプリケーション層30がこの順に階層構造的に体系化されてなる従来の構成に、制御層20から見て各アプリ31～35と同層の位置に外部APIアプリ36を追加すれば良く、基本的に現存のアプリをそのまま使用できる（変更を加えなくても済む。）。これにより、MFPの開発者側にとっては、これらアプリのプログラム変更、確認作業等に手をとられる必要がなく、開発期間を大幅に短縮できると共に開発負担を軽減できる。

【0047】

さらに、XMLを用いてコマンド等を記述できる構成にすることにより、外部のソフトウェア開発者にとって大変扱い易い。また、MFP1の開発者側にとっては、XMLを用いることにより、外部から処理要求として受け付けるべきデータの論理構造を独自に定義でき、設計の自由度が広がるという効果がある。

なお、本発明は、MFP等の画像処理装置に限られず、上記のような処理をコンピュータが実行するためのプログラムであるとしてもよい。また、本発明に係るプログラムは、例えば磁気テープ、フレキシブルディスク等の磁気ディスク、DVD、CD-ROM、CD-R、MO、PDなどの光記録媒体、Smart Media（登録商標）、COMPACTFLASH（登録商標）などのフラッシュメモリ系記録媒体等、コンピュータ読み取り可能な各種記録媒体に記録することが可能であり、当該記録媒体の形態で生産、譲渡等がなされる場合もあるし、プログラムの形態で、インターネットを含む有線、無線の各種ネットワーク、放送、電気通信回線、衛星通信等を介して伝送、供給される場合もある。

【0048】

（変形例）

以上、本発明を実施の形態に基づいて説明してきたが、本発明は、上述の実施の形態に限定されないのは勿論であり、以下のような変形例が考えられる。

（1）上記実施の形態では、画像処理に関する処理要求をXMLを用いて表す構成としたが、これに限られることはなく、例えばHTML（Hypertext Markup Language）等を用いることもできる。この場合、PC50からMFP1に送られる画像処理に関する処理要求を示すデータが、その言語で記述されたデータになるので、外部APIアプリ36は、そのデータを制御層が理解できるデータに変換可能なように構成されることになる。

【0049】

（2）上記実施の形態では、制御層20と外部APIアプリ36を別々のものとして扱ったが、例えば外部APIアプリ36を制御層20の一部として扱う構成とすることもできる。

（3）上記実施の形態では、MFP1とPC50がネットワークを介して接続されたが、上記処理要求等のデータを通信できるものであれば接続方法は限定されず、例えば直接接続される構成とすることもできる。

【0050】

また、本発明をデジタル複写機に適用した例を説明したが、スキャナ、プリンタ、ファクシミリ装置等の画像処理ジョブを実行する画像処理装置一般に適用することができる。

【産業上の利用可能性】

【0051】

本発明にかかる画像処理装置は、自身が備えるアプリでは実行できない画像処理の機能を外部からの指示により実行することが可能であり、複写機、スキャナ、プリンタ等に利用できる。

【図面の簡単な説明】

【0052】

【図1】MFP1のシステム構成を示す図である。

【図2】読み取り解像度を指定したスキャンジョブを実行させるためのコマンド等を含むXMLデータが、PC50からMFP1の外部APIアプリ36を介して制御層20に渡されるまでに変換されて行く様子を例示した模式図である。

【図3】上記スキャンジョブの実行する場合におけるPC50とMFP1間のデータフローを示す模式図である。

【図4】スキャンアプリ33を起動させてスキャンジョブを実行させる場合のデータフローを示した模式図である。

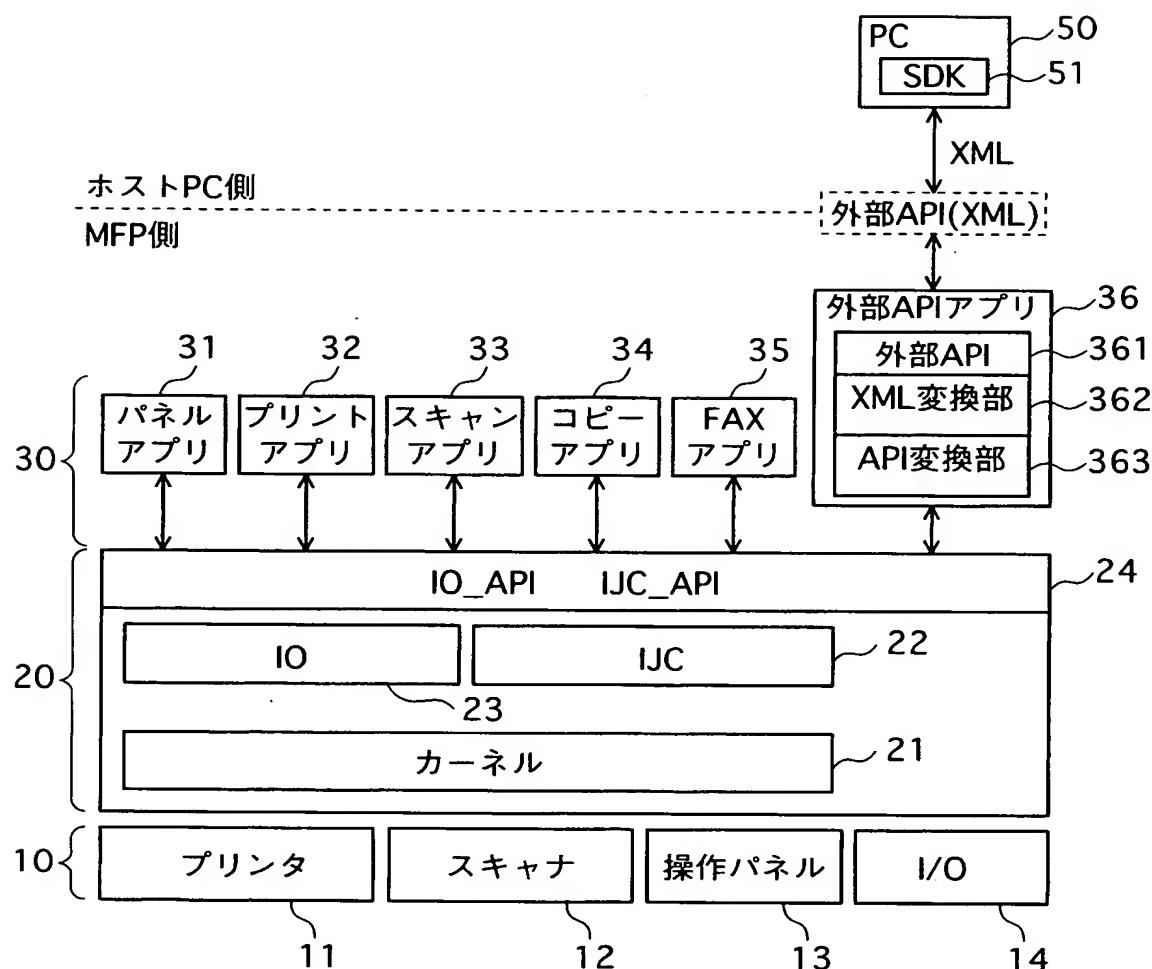
【図5】MFP1が、PC50から送られてくる画像データに基づいてプリントを行うプリントジョブを実行する場合におけるデータフローを示した模式図である。

【符号の説明】

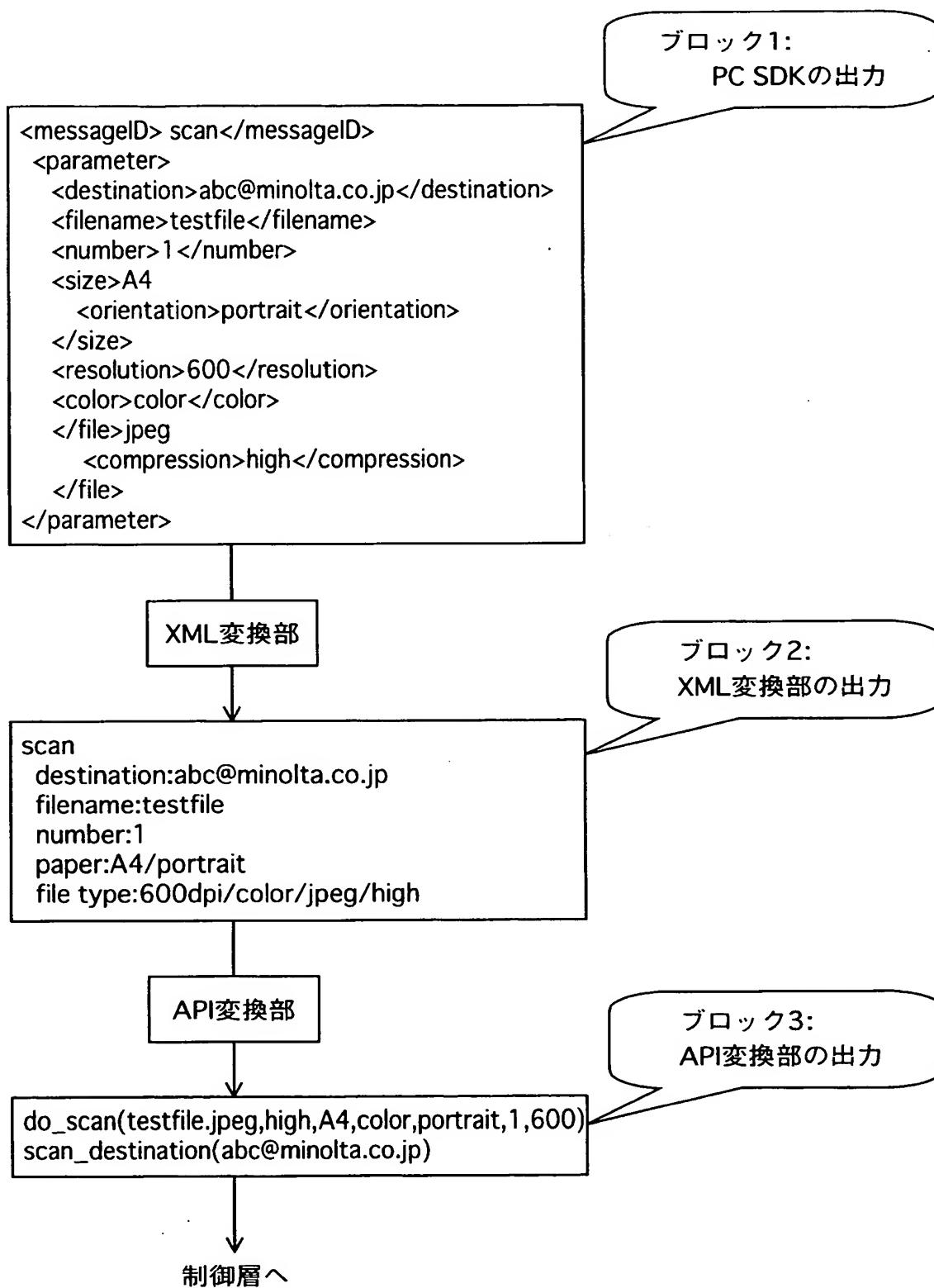
【0053】

1	MFP
11	プリンタ
12	スキャナ
13	操作部
20	制御層
21	カーネル
22	IJC
23	IO
24	IO/IJC API
30	アプリケーション層
32	プリントアプリ
33	スキャンアプリ
36	外部APIアプリ
50	PC
361	外部API
362	XML変換部
363	API変換部

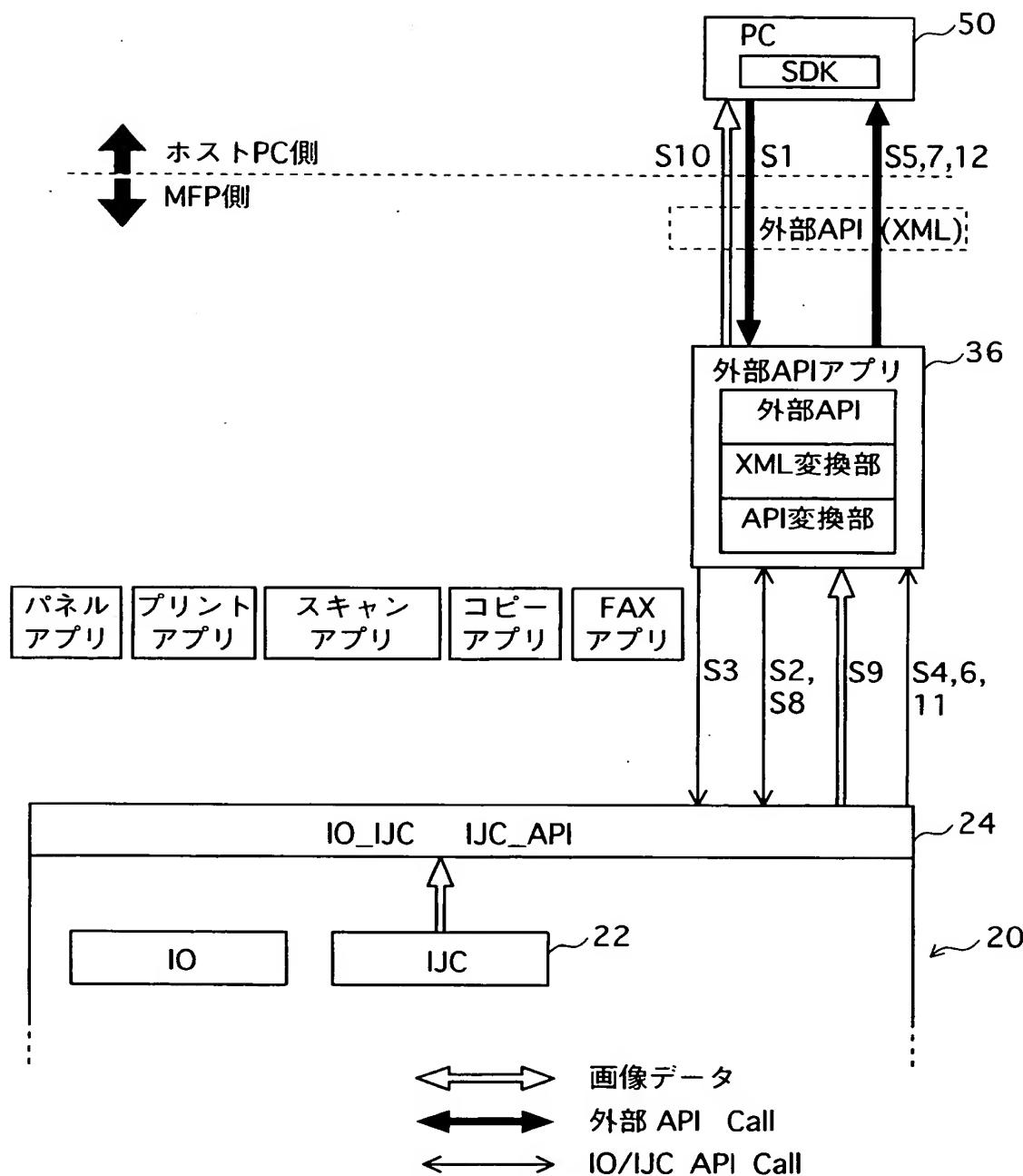
【書類名】 図面
【図1】



【図 2】

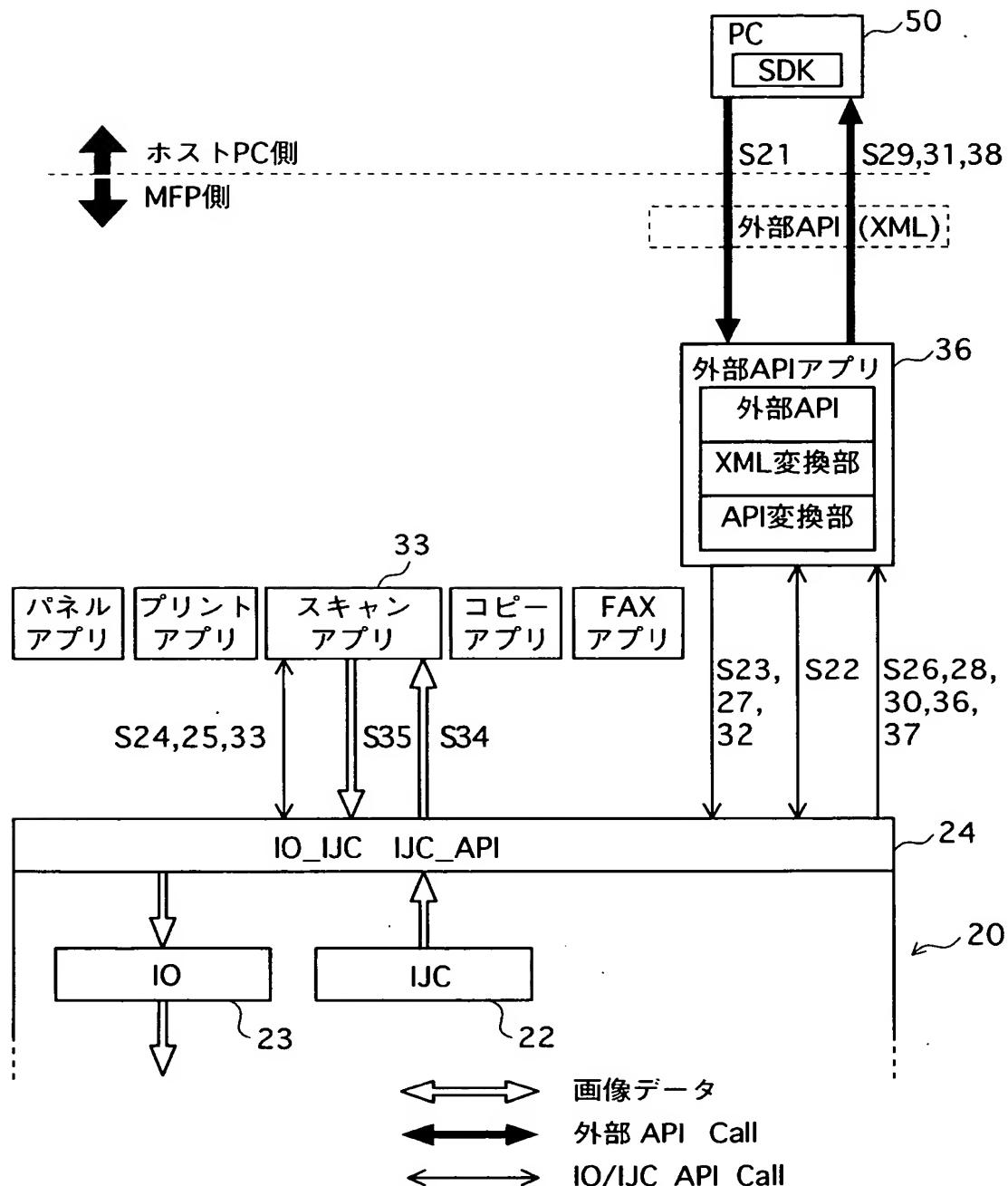


【図3】



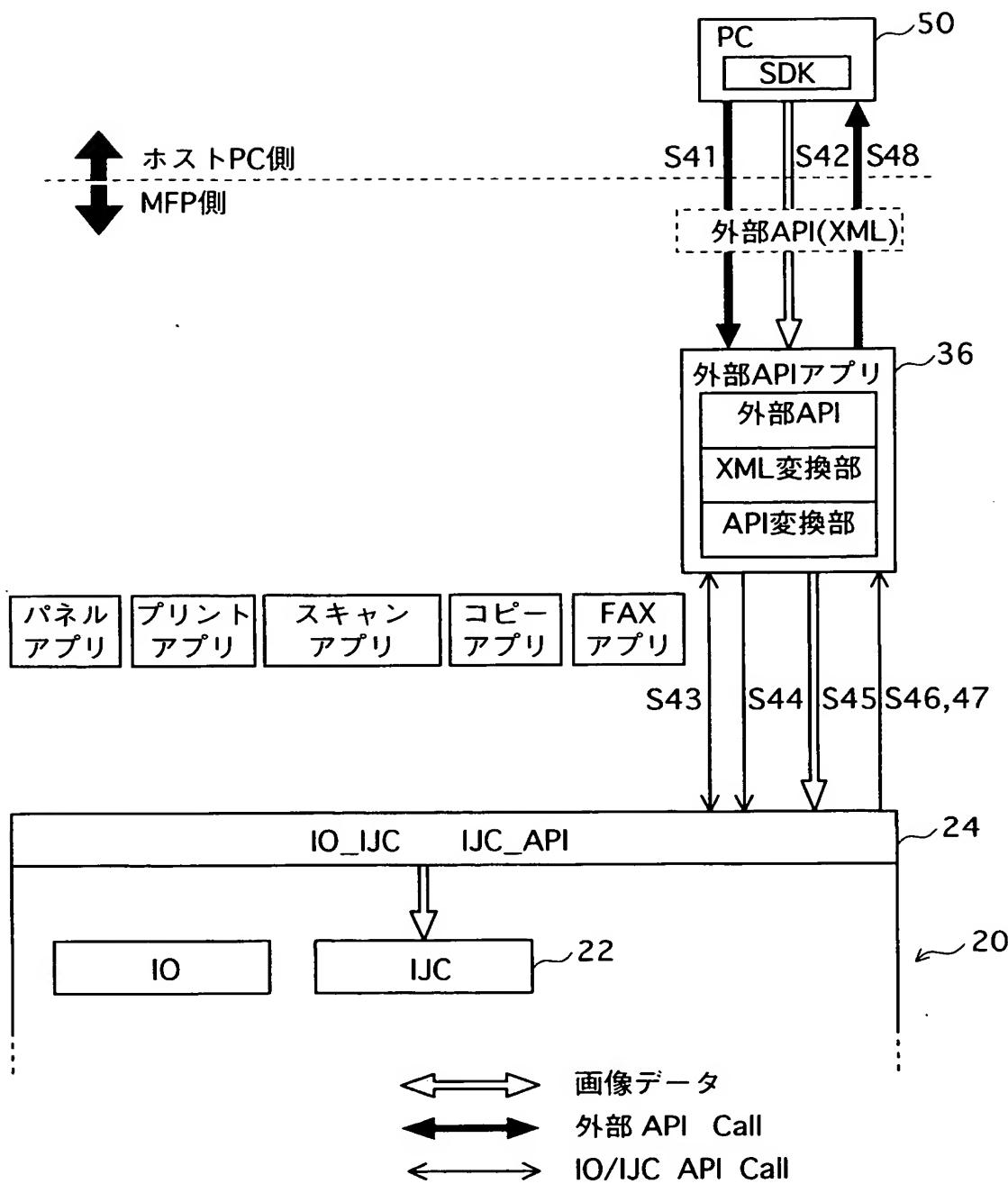
スキャンジョブのデータフロー（スキャン画像をPCに送信）

【図4】



スキャンジョブのデータフロー (PCはスキャン開始指示のみ)

【図5】



プリントジョブのデータフロー (PCからプリントデータを送信)

【書類名】要約書

【要約】

【課題】 機能を容易に拡張でき、その場合に現存のアプリケーションの変更をほとんど必要としない構成の画像処理装置を提供すること。

【解決手段】 デジタル複写機（MFP）は、外部のPC50からの画像処理に関する処理要求を含むXMLデータを受け付けて、それを制御層20のAPI（アプリケーション・プログラム・インターフェース）に対応するコマンドに変換して制御層20に渡す機能を有する外部APIアプリ36を、制御層20から見てアプリケーション層30のアプリケーション31～35と同層の位置に備える。

【選択図】 図1

特願 2003-309293

出願人履歴情報

識別番号 [000006079]

1. 変更年月日 1994年 7月20日
[変更理由] 名称変更
住 所 大阪府大阪市中央区安土町二丁目3番13号 大阪国際ビル
氏 名 ミノルタ株式会社